

Dossier E4

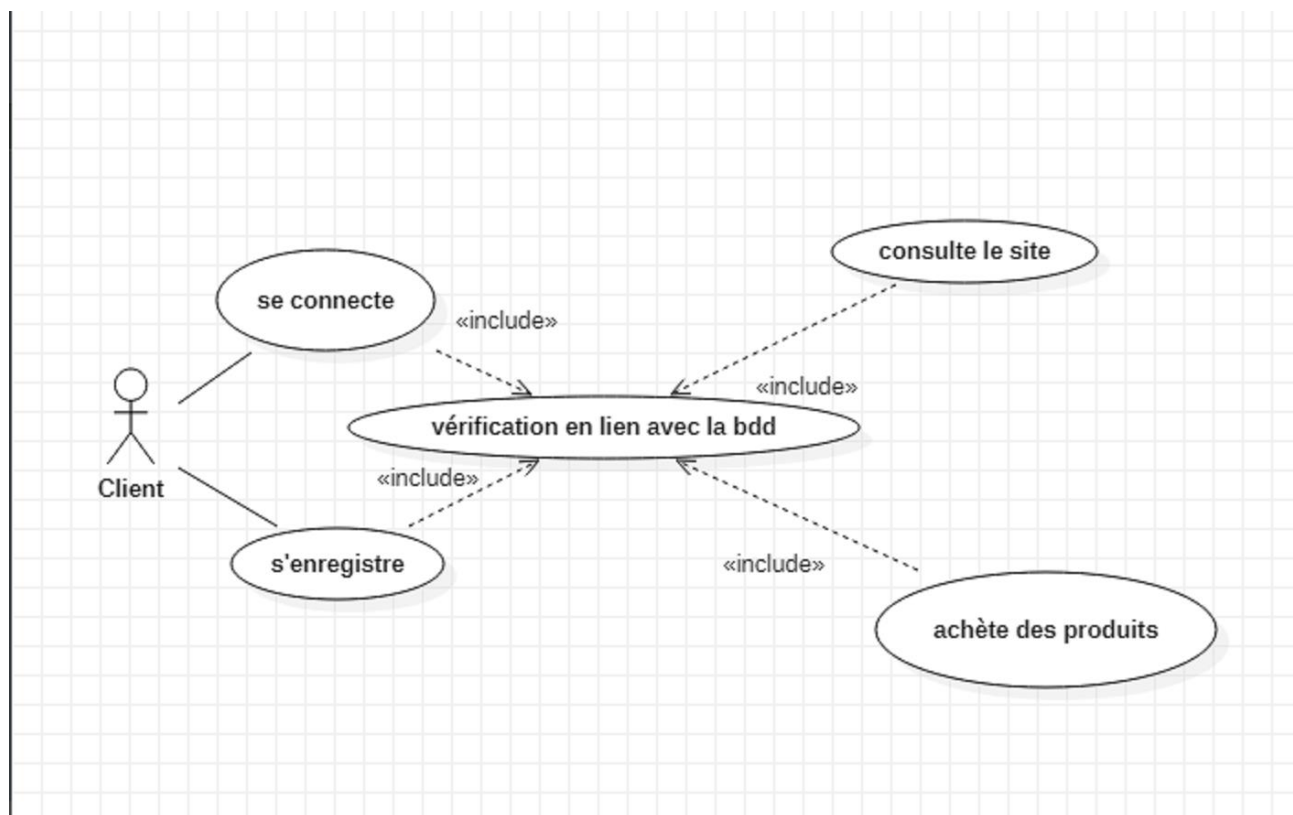
Documentation projet

Le **premier programme** est composé de 3 parties (3 formulaires) :

- La première partie correspond à la **connexion** au site par un visiteur
- La deuxième partie correspond à **l'enregistrement** d'un visiteur sur le site
- La troisième partie correspond au contenu du site lui-même c'est-à-dire la table contenant les produits (informations sur les vêtements).

Ces 2 programmes partageront la même base de données '**achatenligne**'.

La base de données est composée de 2 tables '**connexions**' et '**stock**'.



Le schéma est un use case diagram est une représentation du comportement fonctionnel d'un système logiciel c'est entre guillemet ce que le client pourra faire.

Tout d'abord je devais créer la base de données au nom de 'achatenligne' avec comme table 'connexions' ainsi que la table 'stock' n'oublions pas les clés primaires avec les ID. Avec toutes les données qui sont indiquées dans l'énoncé j'ai rempli la table qui peut se faire de façon manuelle comme en SQL.

Pour ce faire j'ai commencé par la page 'formulaireLiens.html' un premier formulaire permettra d'afficher 2 liens HTML l'un pour se *connecter* le deuxième pour *s'enregistrer*.

Voici l'affichage qu'il devra avoir juste en dessous.

Si on choisit de s'enregistrer on arrive sur le formulaire suivant de nom 'enregistrement.html' et si on veut se connecter on arrive sur le formulaire suivant : 'connexion.html'.

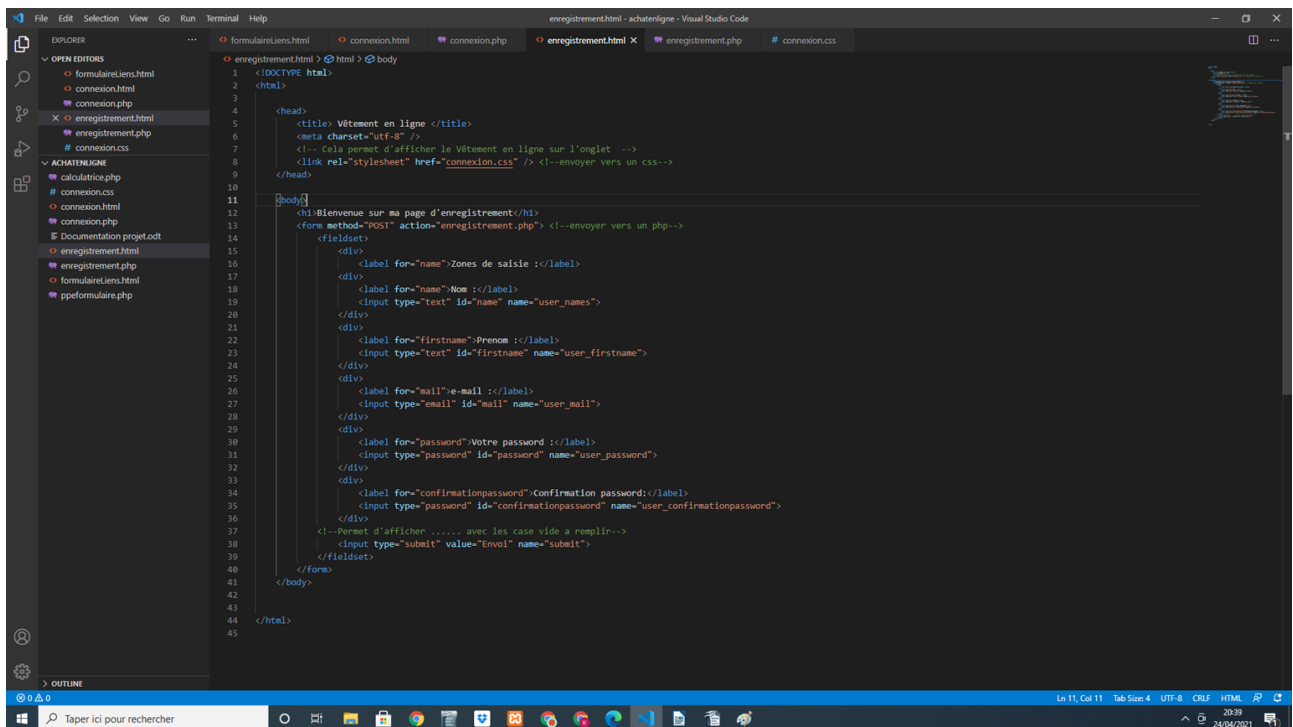
Je vais vous présenter tout ça juste en dessous de la photo ci-dessous. J'ai rajouté le CSS pour la mise en page. Voici la photo avant d'y avoir intégré le CSS et juste après avoir mis le CSS.

```
<link rel="stylesheet" href="connexion.css" /> <!--envoyer vers un css-->
```

Le code juste au-dessus permet de faire la relation avec le fichier formulaireLiens.html avec le fichier connexions.css. Le fieldset permettra de cibler cette partie directement avec la page CSS.

Il y a aussi sur cette page le code permettant que lorsqu'on appuie sur se connecter ou s'identifier qu'elle puissent nous ramener sur la page concernée.

Ensuite je suis passé à la page 'enregistrement.html'. Dans cette page il y a le formulaire avec les cases où nous pouvons remplir nos informations ainsi que le bouton 'envoi' pour envoyer les données, le lien vers le fichier CSS pour la mise en page ainsi que le lien vers le 'enregistrement.php'.



Cette page est bien sur lié au enregistrement.php pour récupéré les données.
Voici l'affichage sur le internet.

Dans le fichier il y a aussi la balise CSS <style> pour pouvoir faire la mise en place de la page.

En premier temps nous vérifions si les champs ne sont pas vides ensuite si c'est vide elle affiche 'Veuillez saisir tous les champs'.

Si c'est bon on entre la requête pour ce connecter a la base de données ensuite ,il y a une deuxième requête pour récupéré les données saisie et les inséré dans la base de donnée avec un 'INSERT' .

En mettant un catch(Exception \$e), on attrape tous les types d'exceptions.

Si c'est bon elle affichera 'donnée inséré dans la base de données'.

Juste après je suis passé a la partie 'connexion.html' .

Cette page permet de ce connecter a son compte dedans il y a le lien vers le fichier 'connexion.css' et vers connexion.php ,le formulaire et le bouton 'envoi'.

Voici comment elle s'affiche sur le web.

Cette page est bien sur lié au connexion.php pour récupéré les données.

Dans la page connexion.php j'ai d'abord commencé a faire la connexion a la base de données avec la requete suivantes :

```
$pdo = new PDO("mysql:host=localhost;dbname=achatenligne","root","");  
echo "<br>Connexion BDD OK";
```

le echo permet d'afficher en php .

```
$genre = $pdo->prepare("select distinct * from stock");  
$executeIsOk= $genre -> execute();  
$genres= $genre -> fetchAll();
```

La requête permet d'afficher toute la table.

Quand on est connecté dans la balise < ?php> elle affiche bon mot de passe.

Dans cette page il y a aussi le test pour vérifier si les mot de passe entré son correct.

En fessant la variable test=false et ensuite en parcourant la table avec un foreach si les informations sont correct la boucle s'arrête et affiche consultation .

Sinon c'est faux avec un `if($test == true)` et la nous n'accéderons pas a notre compte.

Avant cela elle vérifie si les champs sont vide avec un

```
if(((isset($_POST['user_mail']) && isset($_POST['user_password'])) || (isset($_POST['consultation'] )))){
```

Le code affichera aussi le formulaire avec des case de choix.

Exemple :

Ainsi que le bouton 'envoi'.

Quand tout est rempli sur le web, lui le code vérifie aussi si tout les champs sont rempli ,ensuite avec le echo lui affichera ce qui a été choisi.

Il parcourt la table avec un foreach pour affich  ce qui a été choisi grâce au variable ,le tableau est affiché grâce a la balise table.

Balise de fermeture obligatoire la balise TR est utilisée à l'int rieur de l' l ment <table> (balise de type <bloc>). Elle d termine une nouvelle ligne dans celle-ci.

```
/*requete pour rechercher les correspondances entre les champs de la BDD et entrer dans le formulaire "connexion" */
```

```
//On v rifie si tout les champs sont remplis (empty signifie contraire de )
```

```
if(isset($_POST["consultation"])){  
    $quantite=$_POST["quantite"];
```

```
// Requete INSERT en mode prepare
foreach ($stocks as $squan):
    echo $squan['quantite'];
endforeach;
```

```
}
```

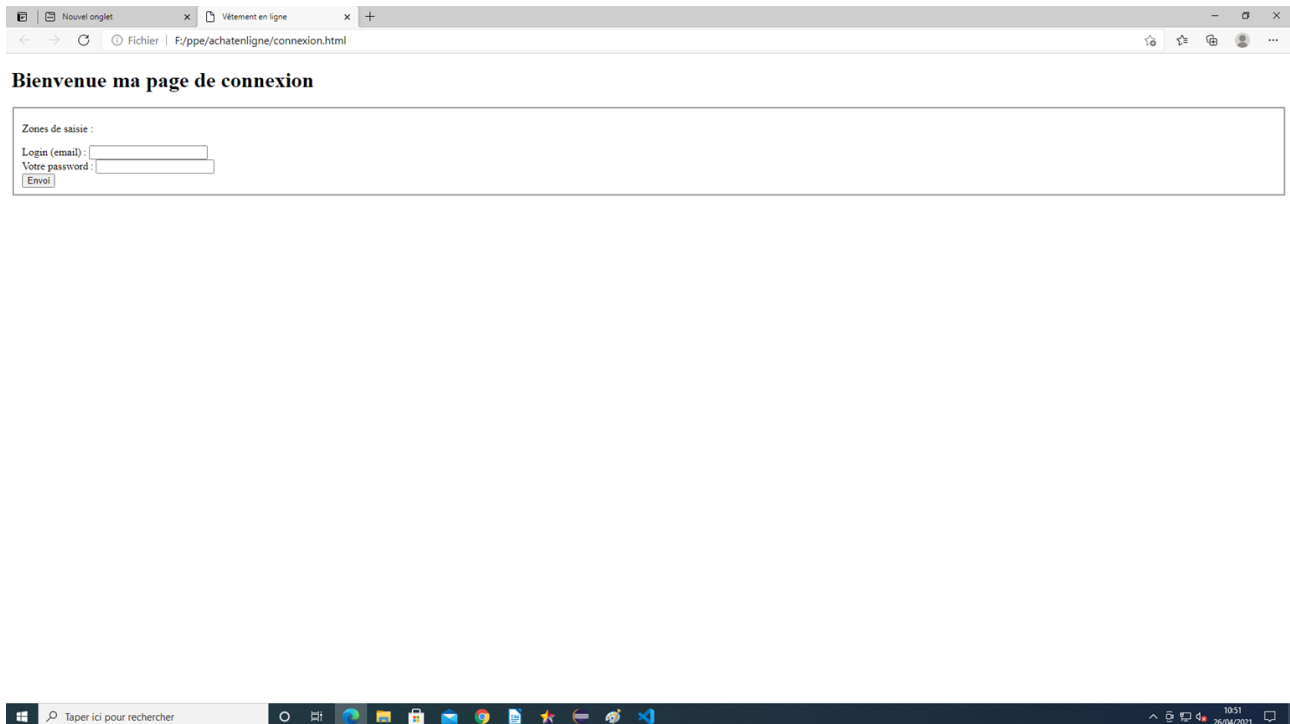
```
>
```

```
1 // connexion a la BDD fac
2 $pdo = new PDO("mysql:host=localhost;dbname=achatenligne","root","");
3 echo "<br>Connexion BDD OK";
4 $genre = $pdo->prepare("select distinct * from stock");
5 $executeIsOk = $genre -> execute();
6 $genres = $genre -> fetchAll();
7
8 $stock = $pdo->prepare("select distinct * from stock");
9 $executeIsOk = $stock -> execute();
10 $stocks = $stock -> fetchAll();
11
12 // (pour le mot de passe)
13 $req=$pdo->prepare("SELECT user_mail, user_password FROM connexions");
14 $executeIsOk = $req->execute(); // execute la requete
15 $data=$req->fetchAll();
16
17 //Verifie si le mot de passe et email sont correct
18
19 if((isset($_POST['user_mail']) && isset($_POST['user_password'])) || (isset($_POST['consultation']))) {
20
21     $test=false;
22     if((isset($_POST['user_mail']) && isset($_POST['user_password']))) {
23
24         foreach($data as $connexion):
25
26             if((($_POST['user_mail']==$connexion['user_mail']) && ($_POST['user_password']==$connexion['user_password']))) {
27
28                 $test=true;
29
30             };
31         endforeach;
32
33     }elseif(isset($_POST["consultation"] )) {
34         $test=true;
35     };
36
37 }
38
39 if($test == true ){
40
41
42
43 }
44
45
46
47
48
49 <html>
```

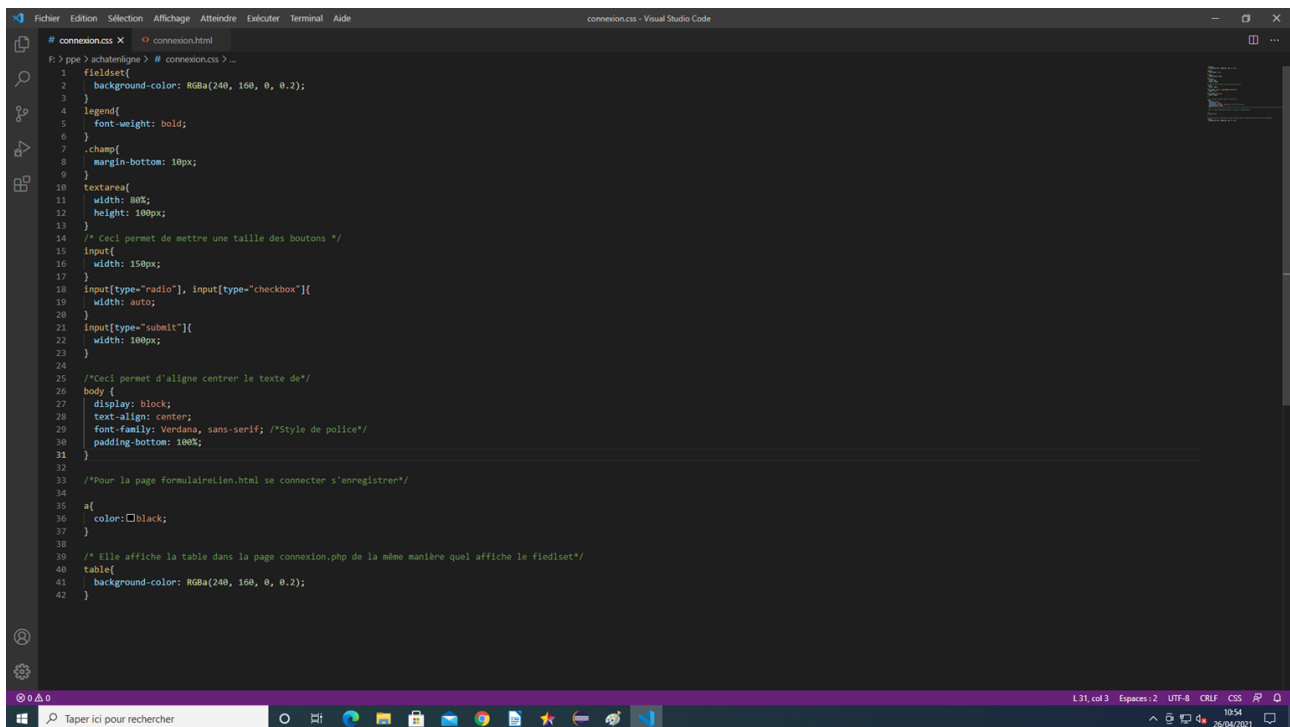
Voici la page ‘connexion.php’ la page ou on pourra effectuer nos achat.

J’ai effectuer le CSS pour la mise en page,dans les screen plus haut le CSS avait déjà été configurer.

Voici un aperçu de la page connexion.html sans le CSS.



Voici un aperçu du code ou on a la configuration du fieldset, la taille des



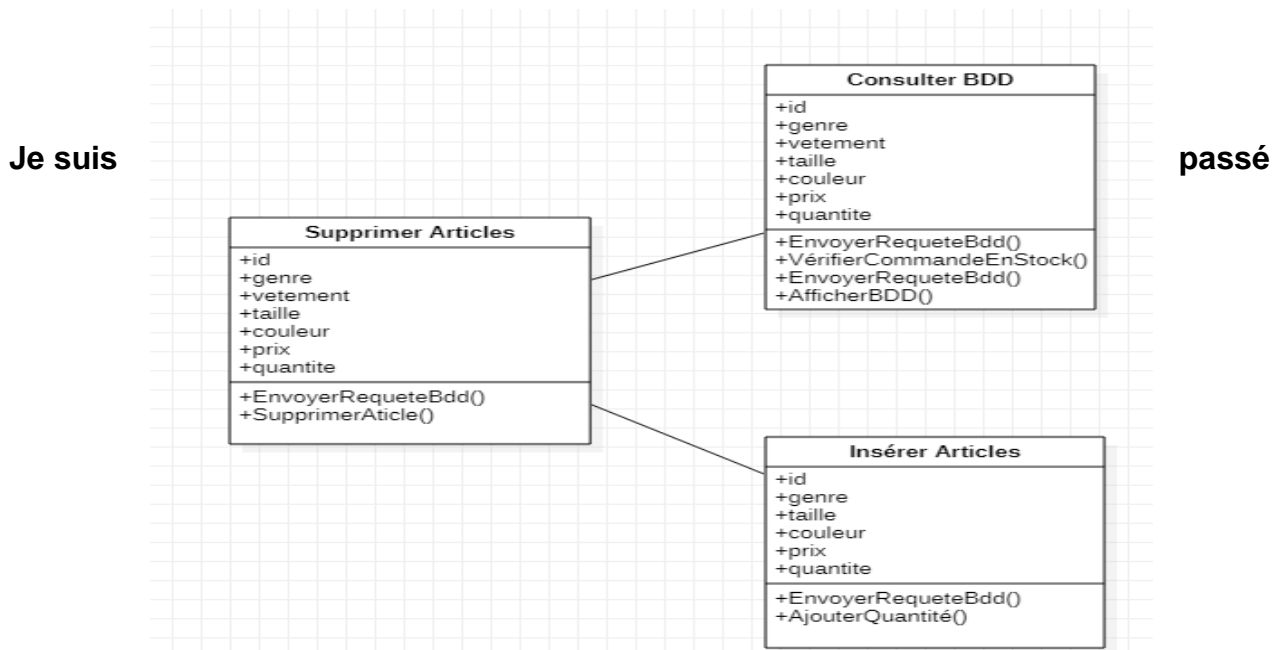
bouton, l'alignement du texte, le texte « s'enregistrer et se connecter » pour la page formulaireLien.html en noir, et affiche la table dans la page connexion.php de la même manière que l'affiche le fieldset.

Partie Java

Le deuxième programme écrit en java sera composé d'une interface graphique (IHM) utilisant les mêmes contrôles que ceux du 1^{er} programme à l'exception des 2 radio boutons, autrement dit, il sera constitué des mêmes 5 menus déroulants. Il aura par contre, 3 boutons supplémentaires :

- Un bouton pour demander l'affichage complète (visualisation) de la table stock
- Un bouton pour insérer de nouveaux articles
- Un bouton pour supprimer des articles de la table

J'ai réaliser tout d'abord un UML pour me facilité le travail avec un diagramme de classe,avec les fonctionnalité de chaque boutons.



maintenant a la partie gérant en créent tout d'abord le fichier 'BddAccess.java'.

Avec comme option dans la class 'BddAccess.java' la recherche de chargement du drivers en mémoire. Elle va afficher ce qui est écrit en parenthèse dans la console,elle se connecte a la BDD.

Il y a aussi dans la page le conteneur de requête la ou les requête son préparer ,après sa elle envoie la requête et récupère la réponse

```

// Recuperation et traitement de la reponse
public void rechercheResultats() {

    //Recuperer de rs les meta data
    //la metadata va chercher le type de structuration de la BDD
    try {
        resMeta = rs.getMetaData();
    } catch (SQLException e1) {

        e1.printStackTrace();
    }
    //recuperer le nombre de colonne de la reponse

    try {
        nbCols = resMeta.getColumnCount();
    } catch (SQLException e) {
  
```



```

        e.printStackTrace();
    }

    //affichage des noms des colonnes
    for(int i = 1; i<= nbCols; i++) {
        try {
            System.out.print(resMeta.getColumnName(i) + " |"+" " );
        } catch (SQLException e) {

            e.printStackTrace();
        }
    }
    System.out.println();
    System.out.println("-----");

    //traitement de la requete
    try {
        while(rs.next()) {
            for(int i = 1; i <= nbCols; i++) {
                System.out.print(rs.getObject(i).toString() + " ");
            }
            System.out.println();
        }
    } catch (SQLException e) {

        e.printStackTrace();
    }
}
}

```

Puis après le fichier 'MainIhm.java'.

Ou il y aura les requête préparer pour pouvoir afficher la table connexions ainsi que la table stocks,et la requête pour ce connecter a la base de données.

Voici la class qui fera apparaître la fenêtre.

Ensuite viens la fenêtre Stock.java ou il y a nos attribut getter et setter.

Ces noms proviennent de « get » (« une méthode pour obtenir la valeur d'un champ ») et « set » (« une méthode pour définir la valeur d'un champ »).

Ici, nous utilisons à la fois des getters et des setters. Tout d'abord, nous utilisons un getter pour obtenir et afficher le nom d'origine des valeur de la table stock. Ensuite, nous utilisons un setter pour lui affecter un nouveau nom dans le champ *getGenre* **par exemple** (« ... »). Ensuite, nous utilisons le getter une nouvelle fois pour obtenir le nom (pour vérifier s'il a vraiment changé).

Voici un aperçu.

Je suis ensuite passé a la class StockSelect.java ou il y a la méthode pour récupérer le stock recherche et chargement du driver en mémoire.

Voici l'application java quand on appuie sur la consultation.

GERANT SITE

Genre(H/F): Genre

vetement: Type Vetement

taille:

Couleurs:

quantite:

prix:

Consultation BDD

Insérer articles

Supprimer articles

Affichages

1	M	Pantalon	Adulte	Beige	15	3
2	F	Robe	Adulte	Bleu	20	4
3	M	Pantalon	7-12ans	Vert	35	7
4	M	Veste	13-18ans	noir	40	5